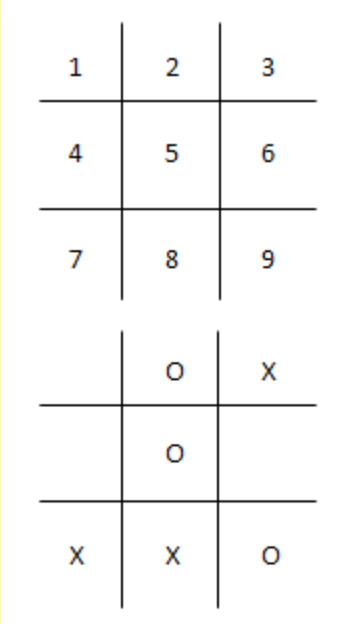
Tic-Tac-Toe: Introduction

Introduction

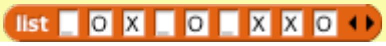
Now that you've learned about higher order functions (HOFs), we're going to use them for a larger example. You are going to make a program that plays a decent game of tic-tac-toe. First of all, you should understand how a mid-game tic-tac-toe board will be represented in the project, as a list of what's in each of the nine squares.



1	2	3
4	5	6
7	8	9

	O	X
	O	
X	X	O

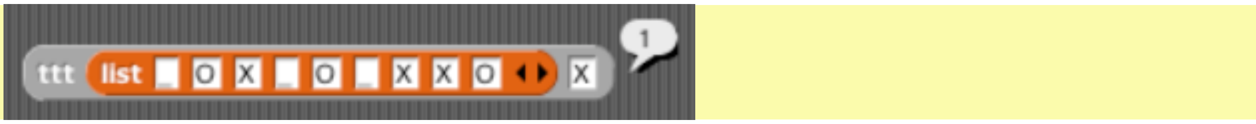
We will represent the board as a list whose items are in the order of the numbers above. In this sample board the first item is an underline (which represents a blank square to make the list more human-readable), and the second item is a O. We will call this the ***position.*** (We'll see shortly that our program will also use a second representation of the board that's easier for the program to manipulate.)



```
list _ O X _ O _ X X O
```

The program will allow a human user to play against the computer. Your goal in this project is to write a block, `ttt`, that will determine the computer's next move based on the current position. (Tic-tac-toe isn't a very challenging game, for people, but as a first game-strategy project it's just right.)

The `ttt` block will report the computer's next move given the current configuration. It takes two inputs: the current position and whether the computer is playing X or O. If the computer is X and using the board above, we can use the `ttt` block like this:



```
ttt (list _ O X _ O _ X X O ◂▸) X
1
```

So the computer will move in the first square, which makes sense.